EXHIBIT A

Intelliforms

owners

Team	Browser
Area	Simplicity
Feature	Intelliforms
Program Manager	Frank Chang
Development	Erik Snapper
Test	Clint Jorgenson
Product Design	Jennifer Shetterly Amy Lee
Usability	Gayna Williams
Current Status	Last edited: [redacted]

Back to Top

summary

Profile Assistant Summary

> The purpose of the Profile Assistant (PA) was to make it easy for users to share registration and demographic information with sites that required this information. The goal was to eliminate the need for users to repeatedly enter information such as their address or e-mail name for each site's registration page. User information was stored securely in protected storage on the client computer. Web servers could request to read this information, but it was shared only if users gave their consent in the PA confirmation dialog box.

Why Intelliforms?

> PA required the web site to write script to request information from the user and furthermore worked only on releases of IE4+. To make this process easier for websites, we would like to be able to offer suggestions on field content by matching on both the field name and on standard identifiers for the form fields. We could then know what information is required and offer the user a way to put that information on the page. Therefore, sites would have to make minor, if any, changes to have Intelliforms work on their site.
>
> The other main problem with PA was that the UI was intrusive to the user and therefore part of the reason that it was rarely used by websites. Instead of popping up a dialog box, the UI for Intelliforms is transparent to the actual page. What we do

with Profile Assistant is in the *Data Sources for the Suggestions* section.

Intelliforms corrects many of the problems associated with Profile Assistant. However, it should not be viewed merely as a replacement for PA. As mentioned above, the purpose of the Profile Assistant (PA) was to make it easy for users to share registration and demographic information with sites that required this information. Intelliforms offers the user assistance in completing all single line edit fields on the web. As a result, Intelliforms also covers the functionality of PA.

Back to Top

guiding principles

- Utilized. It must be easy enough for sites to incorporate Intelliforms that sites will add support for this feature. To that end, we must also make clear to the user that the browser is the one maintaining their data and not websites.
- Useful. The user experience must be enhanced by the help we offer in filling out forms. This means no extensive UI components.
- Perceived and actual privacy. The user must perceive the feature to be keeping his/her data secure. User data must also be safe against physical theft via encryptions of that data.

Back to Top

scenario

Click here to see screen shots of cases where Intelliforms could be used

Back to Top

UI Design

The UI for Intelliforms will be streamlined as to be transparent to users. Each web page with forms on it will look exactly as it does until the user begins to fill out the form. At that point, if that form field matches criterion we have defined, IE will popup an AutoComplete box which is the same one used in the Address Bar. Therefore, the user's mental model of the process of filling out a form remains the same as before, except that we provide a clear way to reduce work. See the spec on AddressBar Improvements for exact behavior of the AutoComplete box.

ISVs and other sites have expressed concern that users might see Intelliforms data as coming from the site rather than from IE itself. This arises from the fact that the UI appears to come from within the page and not from any direct user action on the "IE parts" (menus, toolbar buttons, etc). One way to tell the user that the data is being stored by IE is to strengthen a linkage between Intelliforms and the AddressBar AutoComplete by using the same visual indicators. Taking this a step further, we could simply tell users that IE stores the data by branding the AutoComplete box with IE.

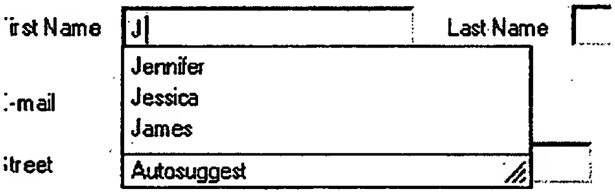
Suggestions:

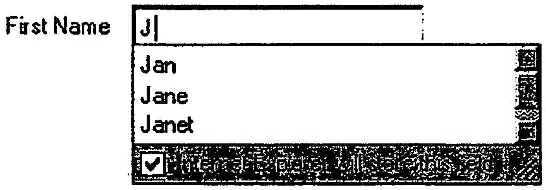
- The short menu in the icon has a "what's this feature..." menu item. That brings up a dialog explaining the feature in greater detail. There can also be an entry point to editing the "Me" entry in the WAB here.

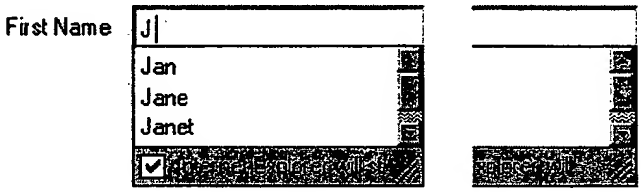
Since the AutoComplete box is the only visible UI for Intelliforms, we have to communicate all relevant data on that box. Conversely, since it takes up such a small screen area, we must be concise in communicating that information. The information we need to convey is:

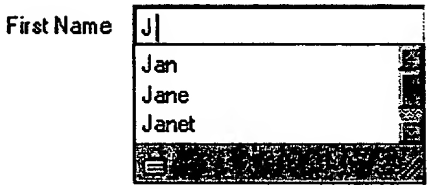
Users should know one of these intuitively...	IE, not the site, is **storing** their information. This means the popup must be visible even if we have no suggestions...	or...IE, not the site, is the one **suggesting** information. The popup will not have to be visible if we have no suggestions.
Users should be able to tell...	Intelliforms has no suggestions but is **on**	Intelliforms is **off**

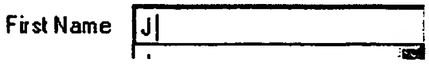
See the pictures below for UI possibilities:

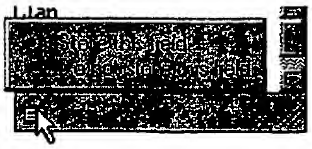




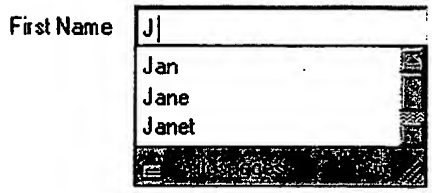








Jan



First Name J|

Jan
Jane
Janet

When will this box appear?

The AutoComplete box will appear only on first character input. A user will then have a chance to read the form, specifically the area that will be obscured by the AutoComplete box, before typing. The AutoComplete box will only show those items that start with user input.

Before the user types anything, he/she can hit the "down" key. This will pop up the AutoComplete box with our suggestions for that field. The user won't have to type anything to fill out a form. For example, the user won't even have to type '9' to enter their zip code of 98052. If we are using the linkages between fields, then we can show what we think the current field should be without the user typing anything. A con of this feature is that it becomes that much easier for other users to see everything I've typed in.

Intelliforms will only work on single-line edit boxes because we don't want to match on multi-line fields like email messages. We will popup on edit fields that contain fewer than 2 characters It's marginally useful if the AutoComplete box appears when the box gets focus. However, to be consistent with the scheme of "if Intelliforms is on, then a box will be shown" we should show it.

How do I navigate? (Key and mouse controls)

The user can do a few things to get the box to appear. They can start typing, or click the down arrow to see a list of everything that is remembered for that particular field. If they start typing, only items which match the first letter will appear. The user can also click on the text box with the mouse to cause the dropdown to appear. If the focus is already on the text field, only one click is required, otherwise the user needs to double click. Once the box is shown, the user can either click with the mouse to select an entry, or down arrow to it. To select the item via the keyboard, the user can hit return, or tab. Using tab will also tab to the next field automatically.

Once the user has finished the form, hitting return while in a text box will submit it.

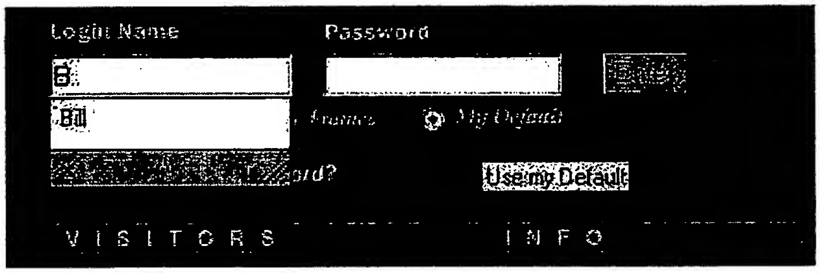
The esc key will make the drop down disappear (if it is shown) and will not fill anything in the textbox, even if an item was highlighted. Typing esc in a textbox will delete the text, but of course, the user can still just hit delete to erase one character at a time.

Here is a chart indicating key action at various states:

		In text box	

	In the text box	dropdown showing	In the dropdown
tab	brings you to the next field	brings you to next field	selects item and brings you to the next field
downarrow	drops the list down	brings you into the list	moves down the list of items
return	submits the form	submits the form	selects item from list and returns you to the textbox, closes the dropdown
escape	deletes the text	dropdown disappears	makes dropdown disappear and returns you to the text box
delete	deletes one character from textbox	deletes one character from text box	deletes that entry from the list permanently

The dropdown can also be removed from the screen by clicking on the X on the bottom bar. This picture illustrates:



What order will the entries be displayed in?

The AutoComplete box should display possible matches to user input in this order:

1. If the name of the field is one of the Common Names or is in the vCard schema and if there is a corresponding entry in our data sources matching the characters entered by the user, display that entry first
2. (Pri2) For each form, we're going to store a whole bunch of (field name, value) pairs in one place - the secondary url data store. Since they're all in one unit, we have a linkage or association between different fields and values. For example, you fill out an ordering form with your name, address, and zip code. Once the user's filled out those fields together on a form, we now know that those different fields are related. So, on another form page with the same fields (name, address, zip code), when the user fills out one of those fields, we have an idea what the other 2 should be. Display any entries for 1 and 2 in alphabetic order.
3. If there are items for 1 or 2, then place a separator bar.
4. The remaining entries mapped to this field name will be listed in alphabetic order.

(may be considered pending usability tests) Run-Once dialog - for first-time users

When the user first installs IE5 and starts typing in a form field, we will bring up a dialog. This dialog will explain what the feature is, emphasizing security and where the data is being stored. This will introduce the user to this feature, which is a new concept to web browsing in many ways. If other new features are doing the same run-once type dialog, we should link to them here.

How to know what information is requested (the Name field)

By looking at forms on the web, it is clear that the contents of the "name" field in form fields of type "text" are often similar when they are looking for the same type of information. For example, when a form is looking for the email address of a user, they often have an input tag:

<input type="text" name="email">

Therefore, if we remember the user-entered value and the name of a particular field, and then encounter a field with the same name, it is likely that the user will want to enter the same value in that field again. This approach allows existing web pages to start using Intelliforms without any changes in HTML.

Whenever IE receives input on a form field with the type="text", it will look for the value of the name field. Using that name value, IE will then look:

1. Into the name indexed data store to see if there were any other completed fields with the same name value. If there are, IE will display those values that begin with the input in the current field in a pop-up AutoComplete box.
2. Into a list of Common Names. These are commonly-used names that we've compiled from research on popular business and e-commerce websites. These names will be mapped to items in the vCard schema, which will be used to prepopulate the AutoComplete Box.
3. Into the defined vCard schema below. If there is data available from sources mapped to that field, then IE will display those values.

vCard Schema:

vCard.Cellular
vCard.Company
vCard.Department
vCard.DisplayName
vCard.Email
vCard.FirstName
vCard.Gender
vCard.Home.City
vCard.Home.Country
vCard.Home.Fax
vCard.Home.Phone
vCard.Home.State
vCard.Home.StreetAddress
vCard.Home.Zipcode
vCard.Homepage
vCard.JobTitle
vCard.LastName
vCard.MiddleName
vCard.Notes
vCard.Office
vCard.Pager
vCard.Business.City
vCard.Business.Country
vCard.Business.Fax
vCard.Business.Phone
vCard.Business.State
vCard.Business.StreetAddress
vCard.Business.URL
vCard.Business.Zipcode

Alternate naming scheme

Few websites currently use the vCard Schema, and getting them to change the names of their textboxes is not all that is involved. The backend to their form processing would have to be altered as well to take into account the new names. We suspect that not many sites will want to do this. We can add yet another attribute to the input tag specifically for intelliforms to identify the field. For example:

<input type="text" name="email" VCARD_NAME="vCard.email">

This way the site can simply add the VCARD_NAME attribute to gain the functionality of intelliforms without going through the extra effort of re-coding their backend.

How will the data be stored?

When the form is submitted, we will store the field name, time, and value submitted into the primary Intelliforms data store indexed by the field name. (pri2) A secondary Intelliforms data store, indexed by URL, is also updated. The data store will be encrypted and stored in the Protected Store. In principle, we should not store any incorrect entries, however, there is no way to detect whether the user has made any mistakes. Therefore, even incorrect form entries will be saved on form submit.

Special cases to this will be discussed in the "Not Remembering Fields" and "Turning on/off" section.

Back to Top

Data sources for the Suggestions

Data will primarily come from the Intelliforms data store in the PStore. When an input field of type="text" is found, we will look in the data store for previously completed fields with the same name. The contents of the AutoComplete box will be from the values corresponding to the field name.

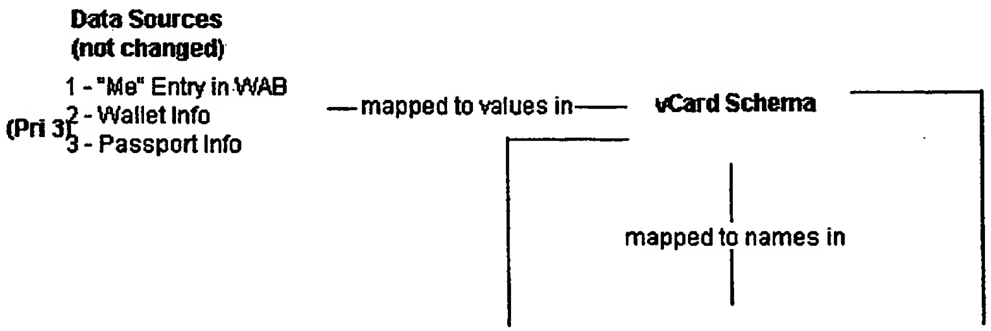
In addition, we will see if the name is mapped to names in the vCard schema. Sources of data that we can map directly to the vCard schema:

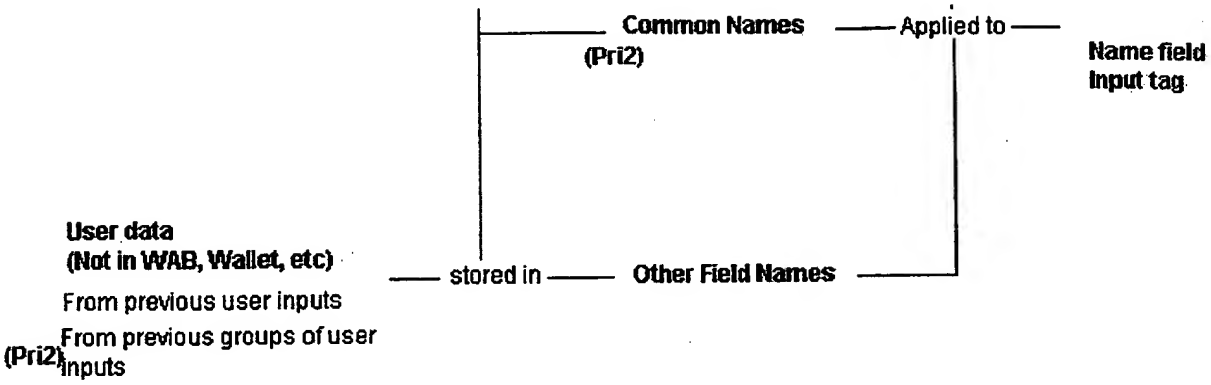
- The "Me" entry of the WAB. It is automatically populated with registration information collected from the user by IE.
- (Pri3) Microsoft Wallet also stores the address and name info for user's credit cards.
- If the user has authenticated to a Passport hub, we can use the profile information stored on that hub. That data is limited to country, postal code, date of birth, gender, PassportID, MSID, nickname, and occupation.

Since the kind of data stored in these sources (name, contact information, etc) is the most commonly requested kind of information on forms, prepopulating the AutoComplete boxes with this data would help right away. We will accomplish this in 2 ways:

1. vCard schema. We can propose a common naming scheme for fields based on the vCard schema created for Profile Assistant.
2. Common Name list. We can thoroughly research the most popular business and e-commerce sites on the web for their field names. We could then map the names of those fields to the corresponding vCard schema entry..

Here's a diagram of how Intelliforms will work:





User data
(Not in WAB, Wallet, etc)
From previous user inputs
(Pri2) From previous groups of user inputs

stored in

Common Names
(Pri2)

Other Field Names

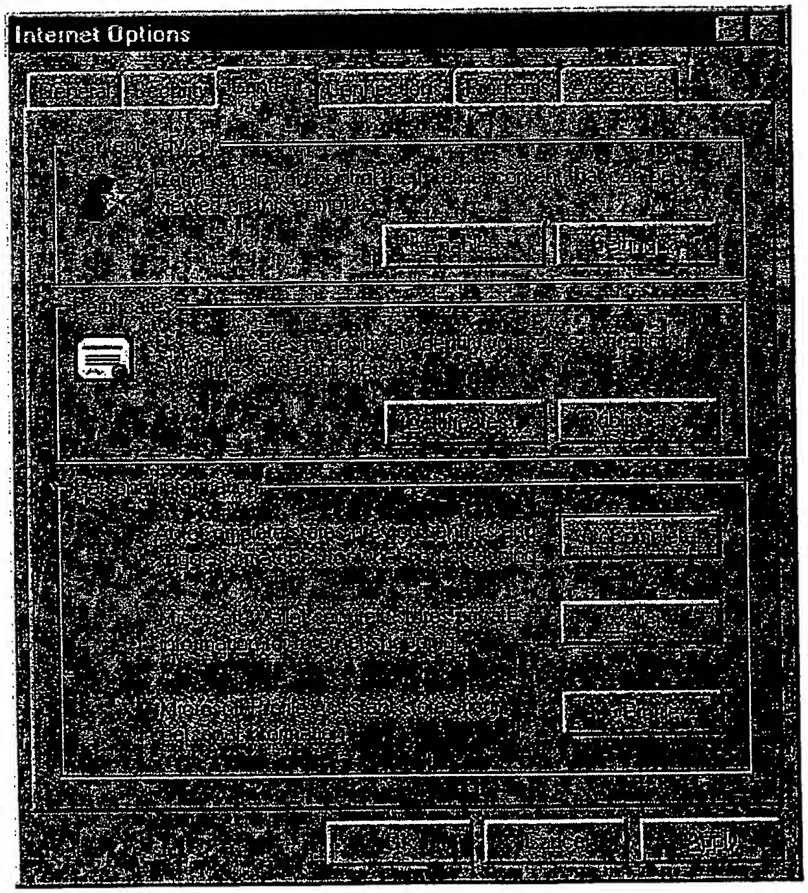
Applied to

Name field
Input tag

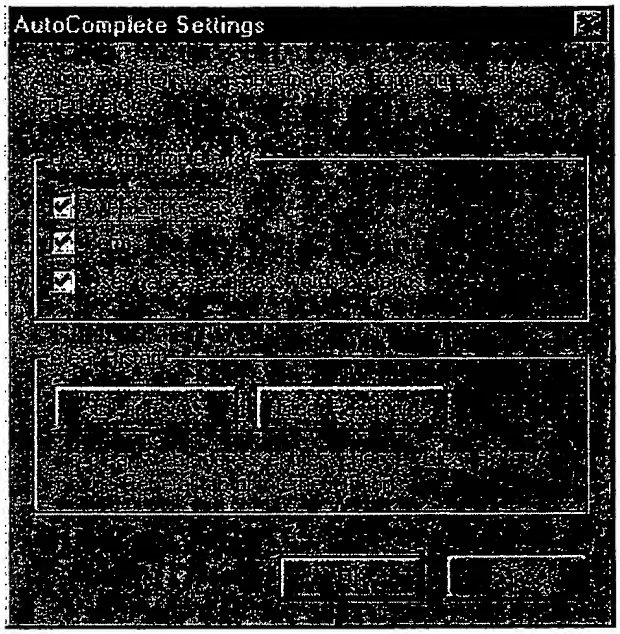
What happens to Profile Assistant? (Pri 2)

As a part of its functionality, Intelliforms will replace Profile Assistant as the way for users to share registration and demographic information with sites that require this information. However, the object model exposed for Profile Assistant will remain for backwards compatibility, so sites that have support for PA will still work. We will remove references to PA in the Internet Options->Contents tab. We will leave a way for the user to edit the "Me" entry of the Windows Address Book. That WAB data will be used to prepopulate the fields in the vCard schema it is matched to.

Here is a picture of what the Content tab will look like.



When the user clicks on the "AutoComplete..." button, this shows up:



discoverability/accessibility

See AddressBar specification.

Back to Top

Security and Privacy

As with any feature that remembers user input, there are security issues with Intelliforms. The issues are:

1. Make sure sites can't collect the stored user data.
2. Make it difficult for outsiders to get at saved user data
3. Ensure the users will perceive that their data is secure

The security measures taken for the first issue are that the site will have no way to see the information stored by IE. The user will see that data in the AutoComplete box, but the actual data won't be put in the form field until the user takes action. The site will have no way to look into this box. The only way that a site might be able to see the data would be for it to change key properties (e.g. tab = down arrow), which can not be done. (NOTE: As of now, testing has found no way of altering key properties to grab intelliforms information. However, if they do, fixing the security hole is a pri 0 item)

We will make it difficult for someone to get at saved user data by encrypting that data and storing it in the Protected Store. If the user is logged into windows, we will use their password as a key to encrypt their data. If the user has not logged onto windows, then we will use a uniqe key per machine to encrypt their data in the PStore.

We will take several measures to ensure that users will perceive their data as being secure. There are two tiers of security measures; the first will be implemented and tested, the second will be implemented in future versions of IE.

The first security tier (implemented first) is:

- Administrators can retrict the use of Intelliforms via the IEAK. This supercedes both users and sites.
- The user can turn it off/on via the Content Options tab. This supercedes site authority.

The second security tier (may be implemented in future versions of IE) is:

- We'll only store on a per field basis. This approach was attempted, but a good way of implementing was not found.

These measures are elaborated in the following sections.

Turning Intelliforms on/off

Model of Intelliforms operation:

Intelliforms	**On**	**Off**
AutoComplete Box	Always shows up	Never shows up
Store field values	We may or may not remember user input, but we will tell the user which one is going to happen.	Fields are never stored
When it should be on or off...	When some fields on a page will be autocompleted and others are not.	When no fields on a page should be autocompleted.

The difference between "Not Storing" and Turning off Intelliforms is this. When we are "Not Storing" some information and Intelliforms is on, we want to tell users that information. This means we will pop up a form of the AutoComplete box, which is minimal, yet conveys that information. Since we expect Intelliforms to be used, we expect that the majority of fields should be completed.

There are three ways to turn Intelliforms on and off. The administrator can turn it off, the user can do it, and a site can do it.

What the Administrator can do

The adminstrator can turn off Intelliforms through the IEAK. IE5 will show forms as if nothing had changed. There will be no way for either the user or the site to turn Intelliforms back on. See the Administration section for more details.

What the user can do

The very second time a user submits a form, a dialog box will come up informing the user of the feature and asking them if they want to turn it on or not. The reason we want it on the second submit is because a message box already comes up on the first submit warning the user of submit is because a message box already comes up on the first submit warning the user of security issues involved with submitting data over the net. That box has a checkbox "Never show this again" which is checked by default. We do not want two message boxes coming up one after another, thus the second submit.

This message box will also contain a button "More Info..." which will bring up the online help on Intelliforms.

The option of turning on/off Intelliforms and AddressBar AutoComplete should be grouped together in Internet Options. They are separate options (i.e. one can be on without the other). Turning Intelliforms off means that we won't store any user input and the Intelliforms UI will not show up. Once a user has turned off Intelliforms, there is no way for a site to turn it back on.

What the site can do

We will give the site a way to turn off Intelliforms, for that form only, via script. This is for sites that believe Intelliforms is detrimental to their site experience and their users will perceive Intelliforms data as being stored by the site. If Intelliforms was previously on, the user should be informed by IE that the site is turning it off and that it will be on for other forms. This can be done by including <Form AutoComplete=off>. This will disable AutoComplete for the entire form.

Secure Sites (HTTPS)

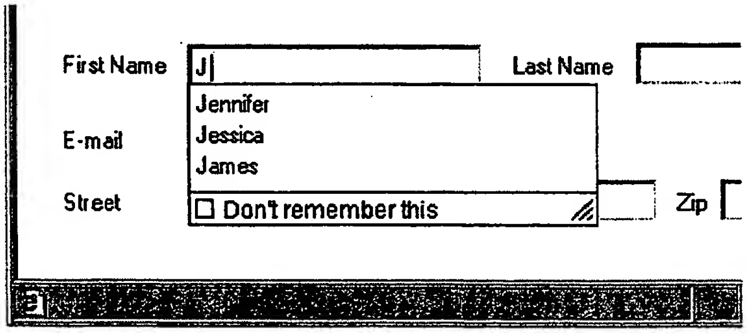
Intelliforms will not work on sites that do not cache information or on HTTPS sites. The following attributes tell Intelliforms to not store information:

pragma:no-cache
cache-control:no-cache
cache-control:no-store
cache-control:private (when not on NT w/per-user cache)

Not storing Intelliforms data (for future versions of IE)

What the user can do - Not storing an individual field

The user may be given the option of not saving an individual field. Here is a picture illustrating one approach:



Another possibility is putting an icon on the AutoComplete box that brought up a short menu. This icon should be a shared visual indicator (with Addressbar) that suggests the box is coming from IE, not the page. The menu would contain:

- Don't store this field - A toggle menu item. Once checked, the AutoComplete box would change to reflect the new state.
- Turn off Intelliforms/AutoComplete - Regular menu item, which turns off Intelliforms globally. The AutoComplete box would disappear.
- Options... - Brings up the Intelliforms Options dialog (if any)

How many individual fields will the user want to not remember? The cons are that the checkbox approach adds another piece of UI every time the user gets an AutoComplete box. The icon/menu approach is less intrusive and is useful in other respects.

The user also has an option in Advanced Options, "Do not save encrypted pages to disk". When this is checked, we will not store any fields by default whenever we go to an encrypted page.

What the site can do

The site can disable AutoComplete for an individual input field through <Input AutoComplete=off>.

(May be considered pending usability tests) What we can do - Not storing numbered input

Any field with numbers will not be remembered, except for those that are named within the vCard schema or those Common Names mapped to the schema. This is because there is a chance they could be a credit card, social security number, bank routing number, or other sensitive data. Numbers probably make up the most sensitive data the user enters in on the web.

Diminished utility: Will Intelliforms be as useful without helping users fill out numbers? Numbers may be the most difficult things to remember and type.

Passwords - Using username field to suggest a password

Intelliforms will autofill-in passwords after a known username is selected from the dropdown. It will remember the login password of each username on a per domain basis.

The feature works by looking at what fields exist on a form. If there is a regular edit field and a single password field, it is probably a login page. Anything with more will not be stored (could be a page to change password). When the user enters a username and password, they will be prompted by the browser to store their password or not. They will be given three choices: Yes, No, No to All (or some variation on this idea). The results of these buttons are:

Yes: Remember my password for this login (based on URL)

No: Do not remember my password for this login. (this is the default action)

No to All: Don't remember my password for any URL and don't show me this window again.

Unless the user clicks on "No to All" the window will appear for each new login encountered. This includes a new user logging in to the same URL. For security reasons, we will not have a "Yes to all" counterpart. "Yes to all" would mean that passwords are stored without warning, and unknowing users would compromise their passwords without even knowing it.

The controls for this feature will be located in the inetcpl with the rest of the intelliforms options. Once the user clicks on "No to all," the only way to turn the feature back on is through the inetcpl.

Changing passwords:

The feature will detect when a different password has been entered for a username already in the database. The user will be prompted if they want this password changed via a dialog like: "The password you entered is different from the one IE has, would you like IE to remember the new password?"

Where is it stored?:

The passwords are stored in the pstore, encrypted along with all the other submitted data. We can not store the passwords in the password list (more secure) because that is only accessible if the user logs in...which most home users do not.

Security issues:

There exist several security concerns with this feature for obvious reasons. This section will first list the concern, then list the solution and/or how it is similar to the "remember my password" feature on MS dialup.

Q. Can a site automatically fill in my username via script and then get my password?

A. NO. A script can not generate key presses, however it CAN change key strokes to make what you type different from what gets to the screen. If this is the case, the user could find themselves transmitting a different login/password combo than they intended. This security hole will be addressed, possibly through trapping window messages before they are sent to Trident.

Q. Can someone spoof a login page and thus get my password? (After I explicitly select my name from the dropdown list?)

A. NO. The feature is URL based. If the URL does not match, firstly no usernames will appear, therefore no passwords will be filled in.

Q. Can someone with physical access to my machine access my password?

A. NO. The password is encrypted in pstore, and when the password auto fills-in, it is still hidden with stars. HOWEVER, any user with access to your machine will be able to login to your accounts by choosing your username from the dropdown. Note that this is if the user does not have their machine configured for MS networking (no logon). If the machine does have a log on, then the passwords are only filled in according to who is logged in. If the user logs out, no passwords can be completed.

logged in. If the user logs out, no passwords can be completed.

Q. Can someone with access to my machine change my password?

A. YES. But it won't do them much good. You can just change it right back. The problem is when the user logs in, and then changes the password via the site. However, this attack will not work if the site requires the user to type in the old password before allowing a new one. This feature will not fill in any password field other than on a login page.

Q. This means anyone with access to my machine can login to any service as me?

A. YES. Just like dialup networking.

Special notes:

- The default setting for any password field, is for IE to **not** store the value.

Commerce pages that request passwords often appear on different URLs. (Pri 2)

We will store a password in the name-indexed data store in pages where the user completes a field with the <input=password> tag. In addition to linking that password to an URL in the data store, we should also link the password and form information to the base URL (i.e. in addition to storing form information for the page https://www.amazon.com/exec/obidos/order2/002-7097885-2828235, we also link to the base URL, https://www.amazon.com. We should remember to look in that url for passwords. Note that we will check to see if the user has decided to not save encrypted pages to the cache (in Advanced Options).

If there are multiple users, how do we prevent one person from using another's password (Pri 2)

Passwords-via-Password: We will require the user to remember only one password (like passport or logging into windows). That password will allow the user access to all the passwords they use while browsing the web. When we first load a page with a tag <input type=password>, then we will pop up a login dialog. Once the user has logged in, the user has been identified and we can now AutoComplete that information. When the user quits the browser, login information is lost and they will have to logon again. This functionality may be provided by Lightweight User Profiles.

An interesting possibility is to use that password to protect all of a user's Intelliforms data. This will involve an ugly dialog, but we can have it pop up only when the user gives focus to a password field. Since sites usually require the user to enter a password before entering secure information, we can be reasonably sure that a user's secure data is hidden from other users.

Handling Credit Cards

(May be considered pending usability tests) How do we deal with websites who wrongly try to collect credit information?

A malicious website could give an input field an innocuous name like "FirstName" while labeling it "Credit Card

Number". An unknowing user may then enter their credit card number, which IE would then save and offer in an AutoComplete box whenever it encounters an input field named "FirstName". This could critically affect the perceived privacy of Intelliforms. To prevent this, we will perform a checksum on field values that are integers. The checksum algorithm is found here (Click here for the source snippets)::

> "For a card with an even number of digits, double every odd numbered digit and subtract 9 if the product is greater than 9. Add up all the even digits as well as the doubled odd digits, and the result must be a multiple of 10 or it's not a valid card. If a card has an odd number of digits, perform the same addition, doubling the even numbered digits instead..."

Since cards can have varying numbers of digits (i.e. Visa has 13 or 16, Amex 15, and MC 16), we will filter the fields on a minimum of 10 digits. This will prevent us from catching zip codes with the checksum. If the field value is a credit card number, then we will not keep a record of the field for the AutoComplete box.

The reason we catch this and not other secure information is that it is something we can clearly catch without cutting out a huge chunk of functionality.

Clearing Intelliforms

Users can clear all intelliforms data by going to Internet Options->Content->AutoComplete->Clear form entries.

The user will also be given the option of clearing the intelliforms data on a set schedule. Just like history clears data every N days, so will intelliforms (if the user chooses to).

Any prepopulated information stored in the "Me" entry of the WAB is not affected.

Users can also delete a specific entry by selecting it in the dropdown and hitting "delete."

Back to Top

setup and administration

The adminstrator can turn off Intelliforms. Forms will look just like they've always looked.

The IEAK already has a policy to disable the Internet Options->Advanced tab. We will have a reg key that represents the checkbox for intelliforms available through inetset.adm. This means that Intelliforms will be available to ISPs/ICPs to disable by default, but only the corporate administrator will be able to not allow users from turning it back on (via inetres.adm > disable advanced tab).

Back to Top

implementation details

Intelliforms is implemented as an HTML behavior attached to FORM elements, so it uses IElementBehavior and IElementBehaviorSite. IAutoComplete is used to interface with the AutoComplete code. For everything else, Intelliforms goes through the Trident object model and event system.

globalization / localization

Back to Top

Future Work (IE 6)

These are things we'd like to get done for IE6. These are not in any particular order.

- *Wallet Integration* - We'd like to incorporate credit card information into Intelliforms. Access should be obtained through Microsoft Wallet's security protocols and storage. If Wallet is not installed, we will use the AutoComplete box, as usual.

  Card data has 4 or 5 parts:cardholder name, brand (generally redundant with the number, but commonly asked for), card number, expiration date, and associated billing address. In addition to all of this, the Wallet associates a friendly name with this collection of information.

  We could have a new standard vcard entry (Wallet.CreditCard, vCard.Wallet, vCard.CreditCard, etc). This means the site is requesting all the card data. The server-side work is such that they'd have to detect whether or not wallet is installed, and show this one field instead of the 10+ individual fields. That's the same as what you had before, so there's no loss there.

  The advantage comes if the page just wants to have fields for all 4-5 pieces of card data and doesn't want to write any server side to detect if wallet is installed. We could define a new schema elements for each piece of data (vCard.Wallet.Name, vCard.Wallet.CardNumber, etc). When the user tabs to any one of those fields, they'd get the dropdown of their cards set up via wallet. They pick one of the cards, and goes through any authentication stuff you guys have. For any other field on that page we suggest all the cards again (in case they want to switch cards), but promote the one they selected so that it's the first suggestion. If they choose another card, we change all the other fields, and do any authentication stuff while telling the user.

- Incorporating all the sources of personal information (Wallet, PA)
- Working on a mouse-based form on interaction where the user would not have to touch the keyboard to complete a form. One idea would be changing the form of an edit field to one similar to drop-down list boxes. The user could then click on the dropdown widget to bring down the AutoComplete box. The box would be populated with previous entries that were entered from most recent to least recent order.
- Working on a method for filling out all fields on a single page.

Back to Top

change history

Date	Changes	By
[redacted]	created	t-rtam
[redacted]	revised security and design	t-rtam
[redacted]	revised security, inetcpl design, and overall design	fchang
[redacted]	revised key controls, added info on auto-passwords	fchang

[redacted]	revised graphics, added info on the first time dialog box	fchang

Back to Top